UNCLASSIFIED

Technical Report
distributed by

DEFENSE TECHNICAL INFORMATION CENTER

DTIC Acquiring Information–Imparting Knowledge

Cameron Station
Alexandria, Virginia 22304-6145

UNCLASSIFIED

Policy on the Redistribution of DTIC-Supplied Information

As a condition for obtaining DTIC services, all information received from DTIC that is not clearly marked for public release will be used only to bid or perform work under a U.S. Government contract or grant or for purposes specifically authorized by the U.S. Government agency that is sponsoring access. Further, the information will not be published for profit or in any manner offered for sale.

Non-compliance may result in termination of access and a requirement to return all information obtained from DTIC.

NOTICE

We are pleased to supply this document in response to your request.

The acquisition of technical reports, notes, memorandums, etc., is an active, ongoing program at the **Defense Technical Information Center (DTIC)** that depends, in part, on the efforts and interest of users and contributors.

Therefore, if you know of the existence of any significant reports, etc., that are not in the **DTIC** collection, we would appreciate receiving copies or information related to their sources and availability.

The appropriate regulations are Department of Defense Directive 3200.12, DoD Scientific and Technical Information Program; Department of Defense Directive 5230.24, Distribution Statements on Technical Documents; American National Standard Institute (ANSI) Standard Z39.18-1987, Scientific and Technical Reports- Organization,Preparation,and Production;Department of Defense 5200.1-R, Information Security Program Regulation.

Our **Programs Management Branch, DTIC-OCP**, will assist in resolving any questions you may have concerning documents to be submitted. Telephone numbers for that office are **(703) 274-6847**, or **DSN 284-6847**. The **Reference Services Branch, DTIC-BCR**, will assist in document identification, ordering and related questions. Telephone numbers for that office are **(703) 274-7633** or **DSN 284-7633**.

DO NOT RETURN THIS DOCUMENT TO DTIC

EACH ACTIVITY IS RESPONSIBLE FOR DESTRUCTION OF THIS DOCUMENT ACCORDING TO APPLICABLE REGULATIONS.

UNCLASSIFIED

MINIMAL-TIME SHIP ROUTING

by

W. E. BLEICK and F. D. FAULKNER

Professors of Mathematics and Mechanics

Research Paper No. 46

UNITED STATES NAVAL POSTGRADUATE SCHOOL

Monterey, California

August 1964

Minimal-Time Ship Routing

V. E. BLEICK and F. D. FAULKNER

U. S. Naval Postgraduate School, Monterey, Calif.

ABSTRACT

A recent theory of minimal-time ship routing through time-dependent ocean wave height and direction fields is put to a numerical test by using a series of semidaily analyses furnished by the U. S. Navy Fleet Numerical Weather Facility. The interpolations and integrations required are found to be feasible. A resume of the theory is given.

1. Introduction

Haltiner, Hamilton and 'Arnason (1962) gave a relaxation method solution to the problem of minimal-time routing of ships through ocean wave height and direction fields dependent on the ship location coordinates only. The theory has been extended by Faulkner (1963) to the case where the wave height and direction depend on time also. The present paper confronts this time-dependent theory with actual wave height and direction analyses from the files of the U. S. Navy Fleet Numerical Weather Facility, and reports on practical problems which had to be solved in a test of the theory.

2. Polar velocity diagram

A basic ingredient of the theory is the polar diagram of Fig. 1, giving the ship velocity v as a function of the angle θ between the ship's heading and the wave direction. A diagram of this kind must be specified for each wave height H. The points L, M and N on the diagram correspond to the ship speed v_h in head waves, v_b in beam waves, and v_f in following waves. Empirical curves for these three speeds as functions of wave height H are available in the pioneer

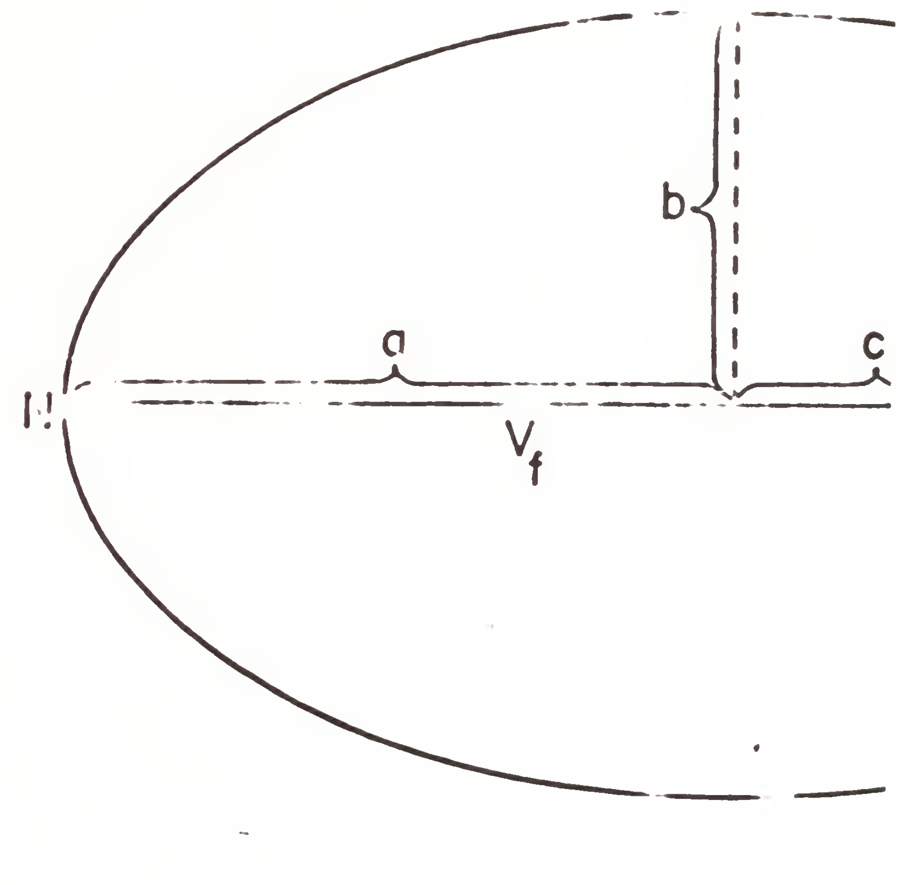
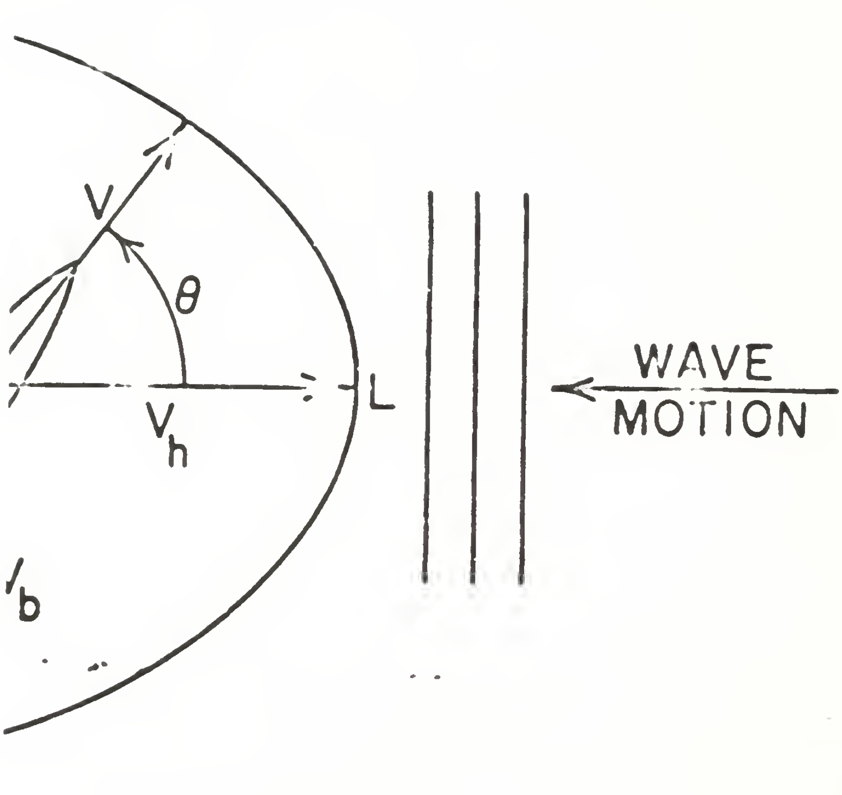


Fig. 1. Polar



/ diagram.

work of James (1957). His P2-S2-R2 ship type curves, shown in Fig. 2, have been chosen for use here. They have the appearance of being arcs of hyperbolas, at least approximately. This was confirmed when a least squares analysis showed that all three of v_h, v_b and v_f can be represented closely as functions of H by the hyperbolic arc

$$c_1+c_2H-[(c_3+c_4H)^2+(c_1-c_0)^2-c_3^2]^{\frac{1}{2}} \quad (1)$$

where c_0 is the point common to all three speeds when H=0. The other four constants are related to the asymptotes $(c_2+c_4)H+c_1+c_3$ and $(c_2-c_4)H+c_1-c_3$. It was decided to construct the polar velocity diagram by fitting an ellipse to the points L, M and N. This resulted in semi principal axes $a=(v_h+v_f)/2$, $b=av_b/(v_hv_f)^{\frac{1}{2}}$, and a distance to the eccentric pole O given by $c=(v_f-v_h)/2$. Note that the pole O is not to be construed as a focus of the ellipse. A further least squares analysis showed that the semiaxes a and b are closely representable also by hyperbolic functions of the form of (1), but not c which must be calculated as $c=(v_f-v_h)/2$.

3. Coordinate system

The ocean wave height and direction data of the semidaily Fleet Numerical Weather Facility analyses are presented in a south-polar stereographic projection of the northern hemisphere upon a plane passing through the circle of 60° North latitude. A rectangular coordinate system is set up in this projecting plane with the Ox and Oy axes parallel to the projections of the meridians of 10° and 100° East longitude respectively. A 62 by 62 grid is constructed using these axes with x=y=31 defining the projection of the North pole. The mesh distance between grid lines corresponds to a distance of 381 km at 60° North latitude where the projection is true. The radius of the equator's projection is 31.205 mesh units. The map scale factor m,

BLANK PAGE

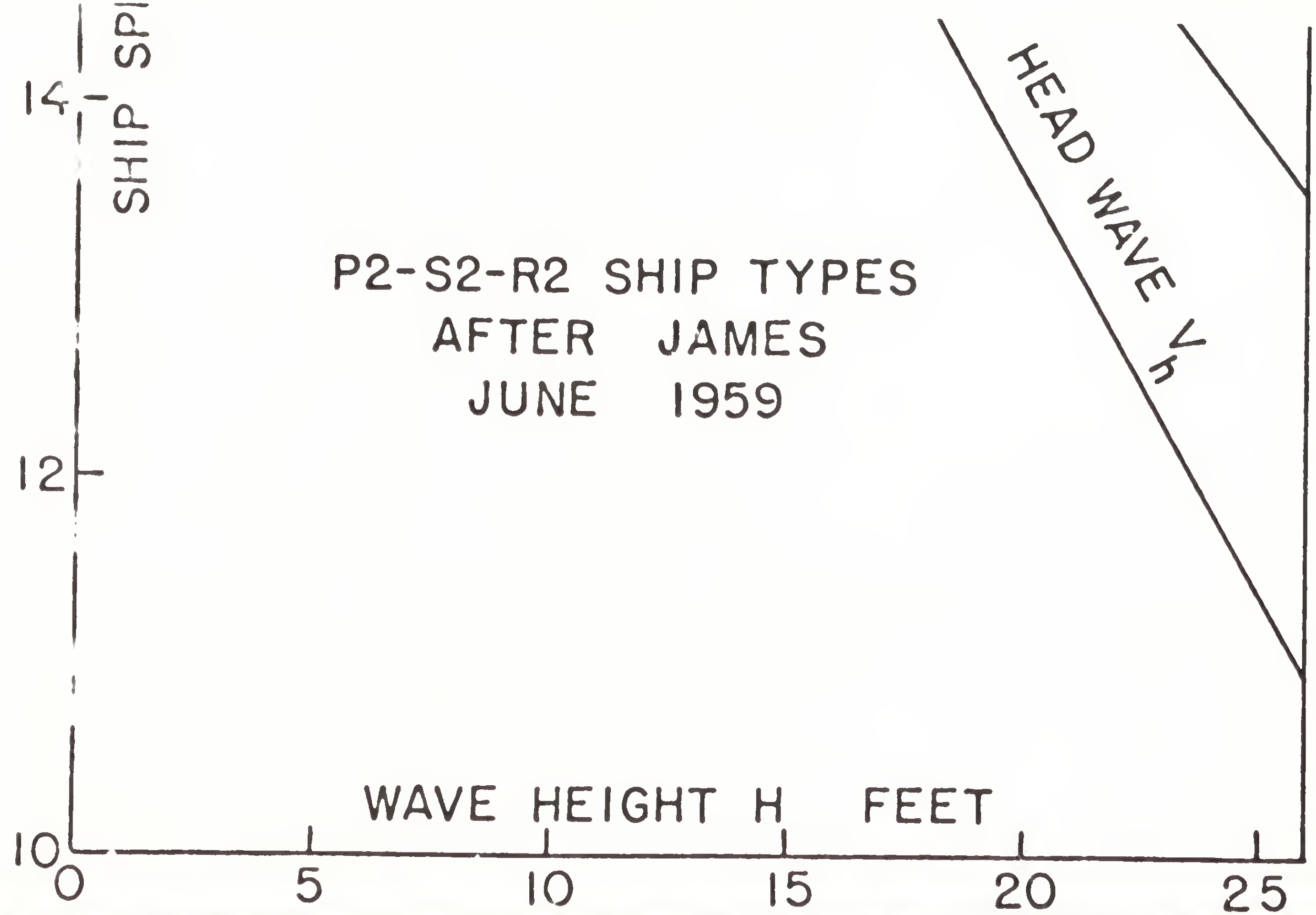
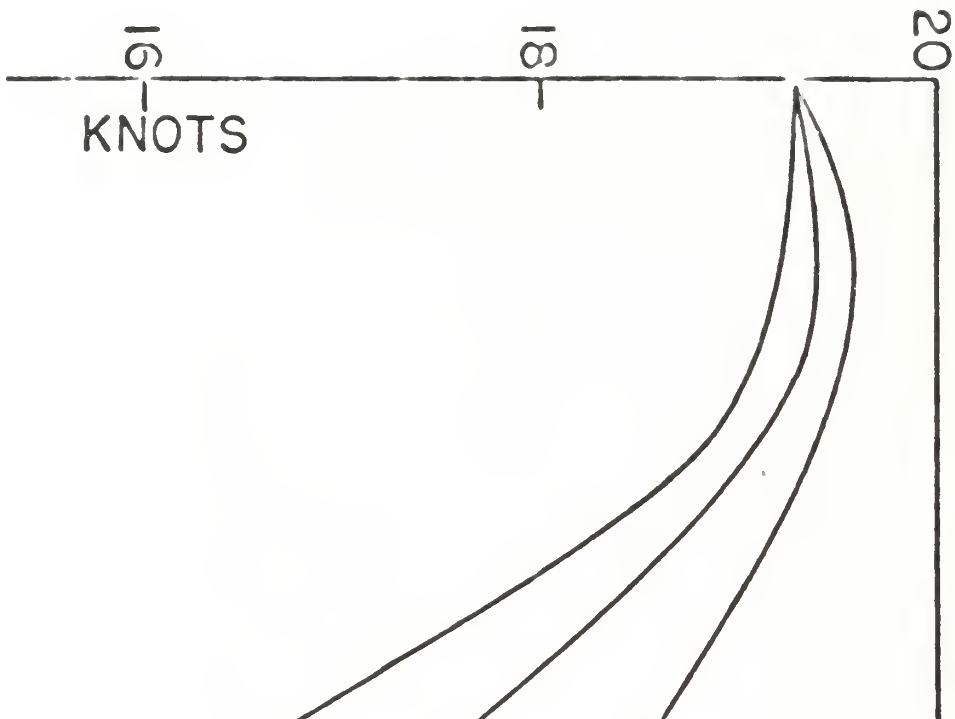


Fig. 2. Ship speed in



g, beam and head waves.

defined as the ratio of a differential distance in the Oxy plane to the corresponding differential distance on the earth's surface, is

$$m=[973.75+(x-31)^2+(y-31)^2]/1043.6 \tag{2}$$

Let x,y be the coordinates of a ship's projection in the Oxy plane at time t. Then the projected speed of the ship is

$$V(x,y,t,\theta)=v(H,\theta)m \tag{3}$$

where $v(H,\theta)$ is the actual geographical ship speed of the polar velocity diagram of Fig. 1, and where $H(x,y,t)$ is obtained by interpolation in the Fleet Numerical Weather Facility grid wave height data. Since the stereographic projection is a conformal transformation preserving angles and their senses, the angle θ is the same in the Oxy plane as on the earth's surface.

4. Resume of the theory

Fig. 3 shows a ship at the point (x,y,t) in the stereographic projection plane on a route from fixed initial point A at t=0 to fixed terminal point B at t=T. The elliptical polar velocity diagram for V=mv is plotted at this point by interpolation in the semidaily wave height $H(x,y,t)$ and wave direction $K(x,y,t)$ grid values of the Fleet Numerical Weather Facility. The special type of interpolation required is discussed in the Appendix. The direction of the ship's velocity vector $\underline{V}=\underline{i}\dot{x}+\underline{j}\dot{y}$ is the control angle p, which is to be chosen at each point so as to minimize the transit time T from A to B. The equations of motion of the ship's projection in the Oxy plane are

$$\varphi_1=\dot{x}-V\cos p=0, \quad \varphi_2=\dot{y}-V\sin p=0 \tag{4}$$

where $V=|\underline{V}|=V(x,y,t,p)$. The problem of minimizing T is equivalent to the Lagrange calculus of variations problem of requiring the integral

$$I = \int_0^T (1+\lambda\varphi_1+\mu\varphi_2)dt \tag{5}$$

BLANK PAGE

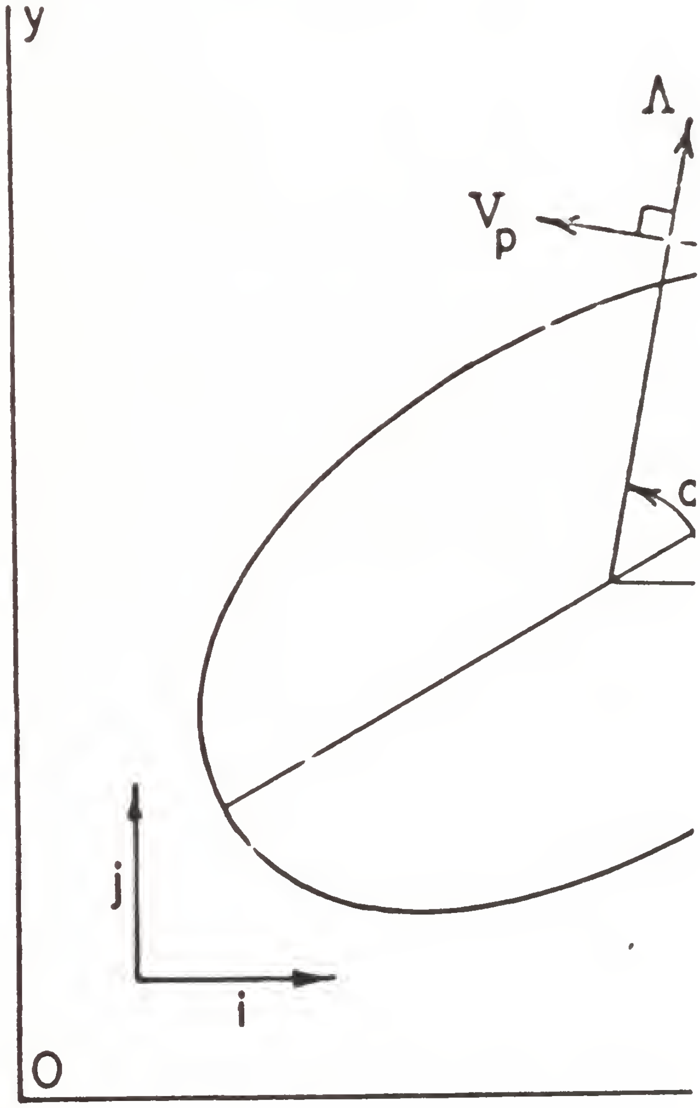
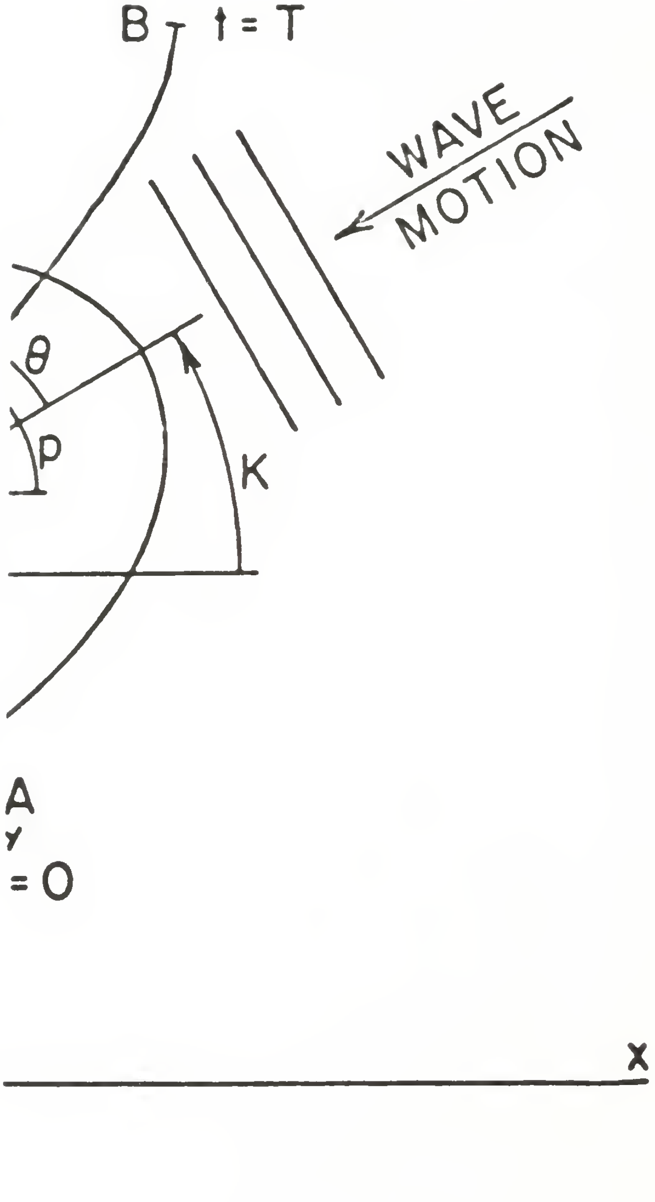


Fig. 3. Ship motior



eographic Oxy plane.

to be stationary, where $\lambda(t)$ and $\mu(t)$ are continua of Lagrangian multipliers. Let the time at the fixed terminal point B be varied to $T+\Delta T$. The vanishing first variation of I is

$$\delta I = \Delta T + [\lambda\delta x + \mu\delta y]_0^T - \int_0^T (\varpi_3\delta x + \varpi_4\delta y + \varpi_5\delta p)dt = 0. \quad (6$$

The coefficients of $\delta x, \delta y, \delta p$ in $\delta I=0$ give the Euler equations (7), (8) and (9), consisting of the adjoint equations

$$\varpi_3 = \dot{\lambda} + (\lambda \cos p + \mu \sin p)V_x = 0, \quad (7$$

$$\varpi_4 = \dot{\mu} + (\lambda \cos p + \mu \sin p)V_y = 0, \quad (8$$

and the scalar product control equation

$$\varpi_5 = \underline{\Lambda} \cdot \underline{V}_p = 0, \quad (9$$

where the adjoint vector $\underline{\Lambda} = \underline{i}\lambda + \underline{j}\mu$, and where $\underline{V}_p = \partial \underline{V}/\partial p = \underline{i}(V_p \cos p - V\sin$ $+\underline{j}(V_p \sin p + V\cos p)$ is the tangent vector to the polar velocity diagr of Fig.3. Eq. (9) implies the orthogonality of $\underline{\Lambda}$ and $\underline{V}_p$ as shown i Fig.3. Eq.(9) may be written also in the form

$$p = \arctan(\mu/\lambda) + \arctan(V_p/V). \quad (10$$

The fixed end points A and B imply that

$$dx(0) = \delta x(0) = 0, \qquad dy(0) = \delta y(0) = 0, \quad (11$$

$$dx(T) = (\dot{x}\Delta t + \delta x)_T = 0, \; dy(T) = (\dot{y}\Delta t + \delta y)_T = 0. \quad (12$$

Use of Eqs.(11) and (12) makes the remaining terms of (6) proportio to ΔT, whose coefficient gives the scalar product transversality condition

$$(\underline{\Lambda} \cdot \underline{V})_T = 1 > 0 \quad (13$$

meaningful for sign only because of the homogeneity of (7) and (8) Eq.(13) implies that the angle $(q-p)$ between $\underline{V}$ and $\underline{\Lambda}$ is acute, as shown in Fig.3. A further implication of (10) and (13) is that the quadrant of p is such that

$$\cos p = (\lambda V - \mu V_p)/\Lambda R, \; \sin p = (\lambda V_p + \mu V)/\Lambda R, \quad (14$$

where $\Lambda=|\underline{\Lambda}|=(\lambda^2+\mu^2)^{\frac{1}{2}}$ and $R=|\underline{V}_p|=(V^2+V_p{}^2)^{\frac{1}{2}}$.

The simultaneous numerical integration of (4), (7), (8) and (14) is carried out together with a Newton-Raphson iteration as follows: Let λ_1,μ_1 and λ_2,μ_2 be two linearly independent solutions of the adjoint Eqs.(7) and (8) corresponding to the columns of the matrix

$$\underline{E}(t)=\begin{bmatrix}\lambda_1 & \lambda_2\\ \mu_1 & \mu_2\end{bmatrix} \tag{15}$$

where $\underline{E}(0)=\underline{I}$ is the identity matrix. The λ,μ of (14) are taken as the linear combinations $\lambda=\lambda_1\cos\alpha+\lambda_2\sin\alpha$ and $\mu=\mu_1\cos\alpha+\mu_2\sin\alpha$. The variation δp is found by total differentiation of (10) to be

$$\delta p=R|\underline{E}|\delta\alpha/\Lambda^2(V^2+2V_p{}^2-VV_{pp}) \tag{16}$$

where $|\underline{E}|$ is the determinant of $\underline{E}$. Assume that a solution of the ship motion Eqs.(4) has been found, corresponding to (7), (8) and (14) for some value of α, which falls short of the fixed end point B at t=T by the coordinate differences $\Delta x(T)$ and $\Delta y(T)$. Using this solution and holding T fixed, find the variation of the vanishing matrix integral

$$\int_0^T[\varphi_1,\varphi_2]\underline{E}(t)dt=0. \tag{17}$$

Since the columns of $\underline{E}(t)$ satisfy the adjoint Eqs.(7) and (8), one obtains the 1×2 matrix equation

$$[\delta x,\delta y]_T\underline{E}(T)=\int_0^T[(V_p\cos p-V\sin p),(V_p\sin p+V\cos p)]\underline{E}\delta p\,dt. \tag{18}$$

Substitution from (14) and (16) into (18) gives

$$[\delta x,\delta y]_T=[-\mu,\lambda]_T J\delta\alpha \tag{19}$$

where

$$J=\frac{1}{|\underline{E}(T)|}\int_0^T R^3|\underline{E}|^2dt/\Lambda^3(V^2+2V_p{}^2-VV_{pp}). \tag{20}$$

Now vary the terminal time from T to T+ΔT and substitute

$$[\delta x,\delta y]_T=[\Delta x,\Delta y]_T-[\dot{x},\dot{y}]_T\Delta T \tag{21}$$

into (19) to obtain the Newton-Raphson equations

$$\begin{aligned} \dot{x}(T)\Delta T - J\mu(T)\delta\alpha &= \Delta x(T) \\ \dot{y}(T)\Delta T + J\lambda(T)\delta\alpha &= \Delta y(T) \end{aligned} \tag{22}$$

for the determination of ΔT and $\delta\alpha$ on a varied trajectory which attempts to correct the errors $\Delta x(T)$ and $\Delta y(T)$. The iteration to successive varied trajectories is continued until the terminal errors are acceptable. A suitable initial guess for the angle α is the inclination angle of the straight line from A to B.

5. Numerical example

Ten successive semidaily analyses of wave height and direction, starting at 06Z on 4 May 1963, were furnished by the Fleet Numerical Weather Facility. The maximum speed of the chosen P2-S2-R2 ship type is 19.6 knots. This combination of data precluded a trip of great length. It was decided to select an area of continued extreme wave height for the example. Such an area was found centering at 30° North latitude and 162° East longitude. Figs. 4, 5 and 6 show two computed minimal-time ship tracks in the area, with contours of wave height in feet and wave direction arrows. The arc traversed by the ship during the 6 hours preceding and/or the 6 hours following the time of each Figure is shown as a dashed curve. The minimal-time track AB required 2.488 days with a 3.0% saving over the geodesic track. The minimal-time track CD required 2.656 days with a 1.3% saving over the geodesic track. The severity of the sea conditions in the area preclude any more spectacular saving. The highly non-analytic nature of the wave height in the area was found to affect the convergence of the Newton-Raphson iteration of (22). It was found necessary to halve the values of ΔT and $\delta\alpha$ in order to avoid a divergent oscillation. Resort to this stratagem was found to be unneces-

BLANK PAGE

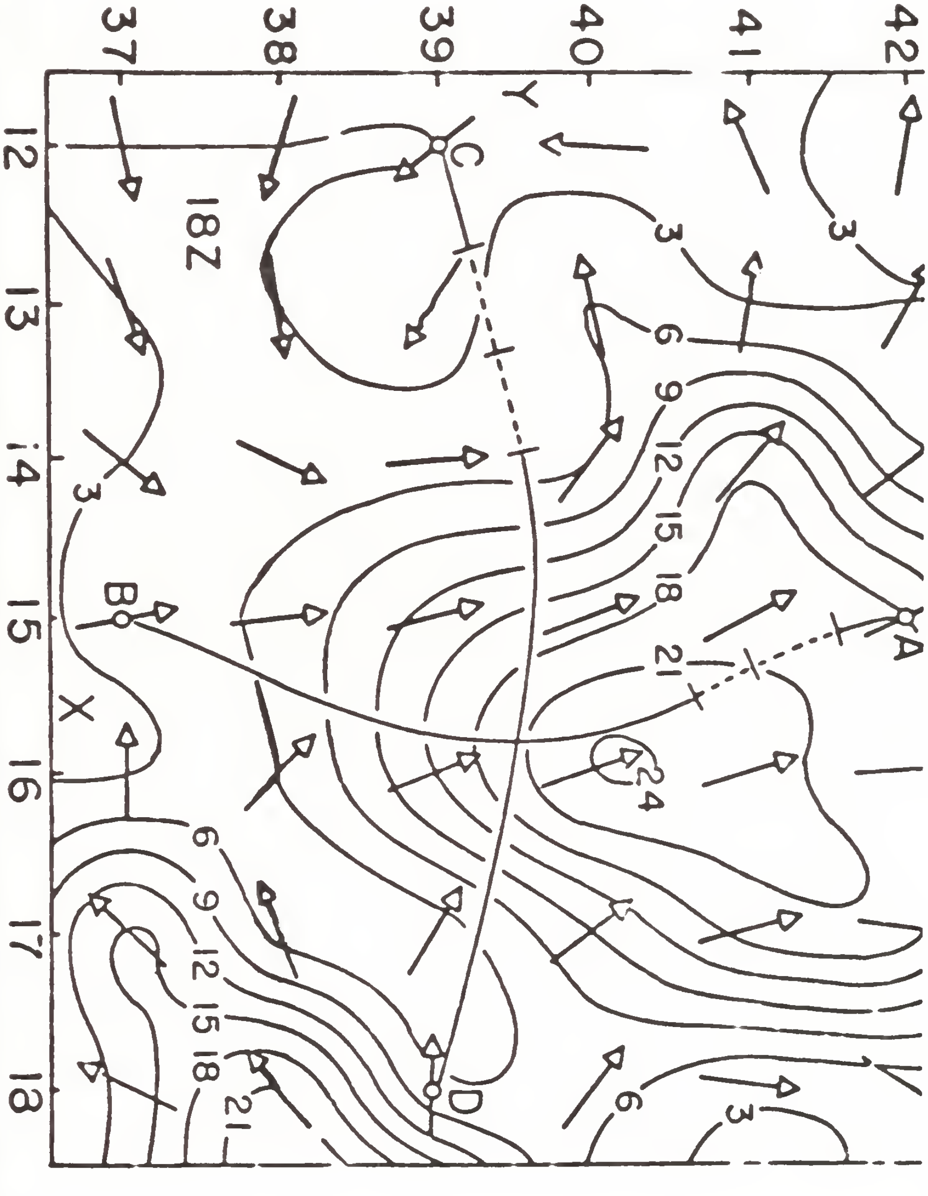
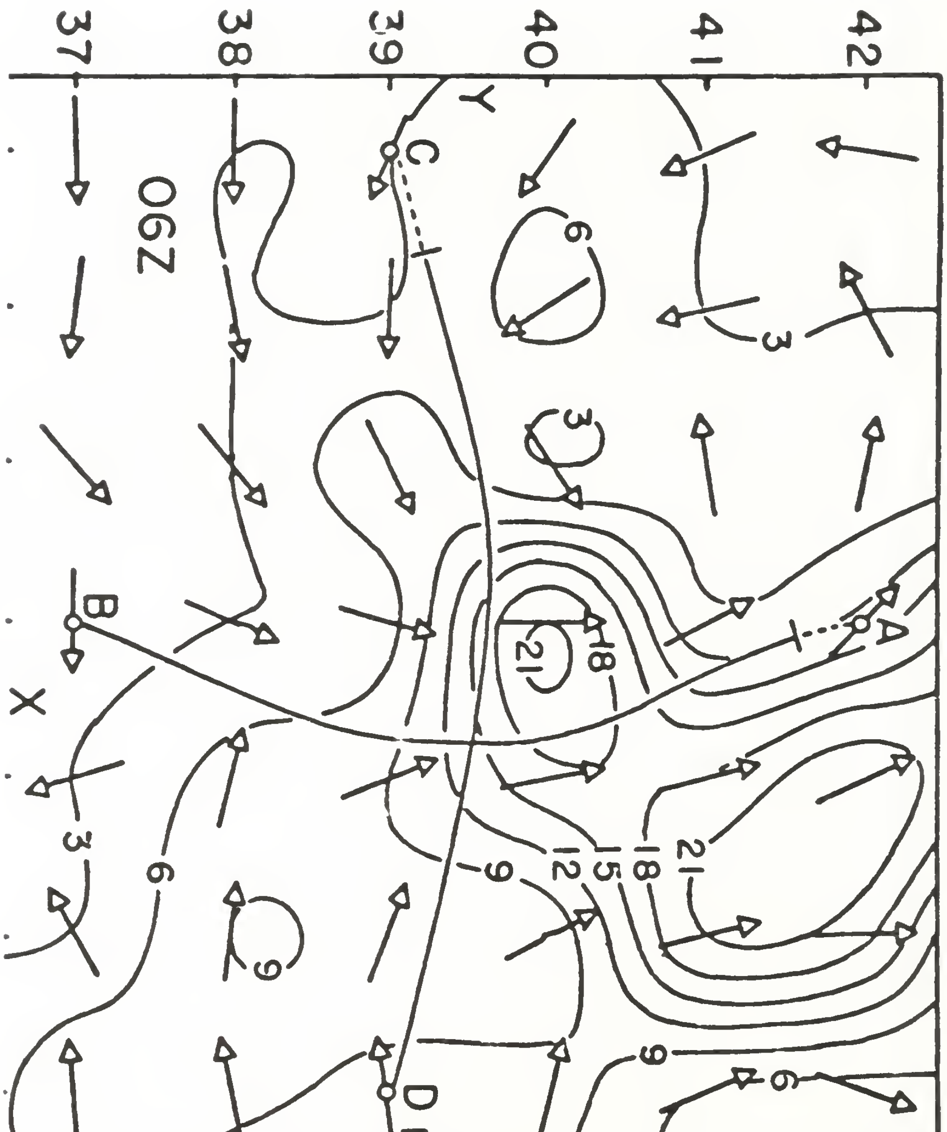


Fig. 4. Sea conditions



nd 18Z on 4 May 1963.

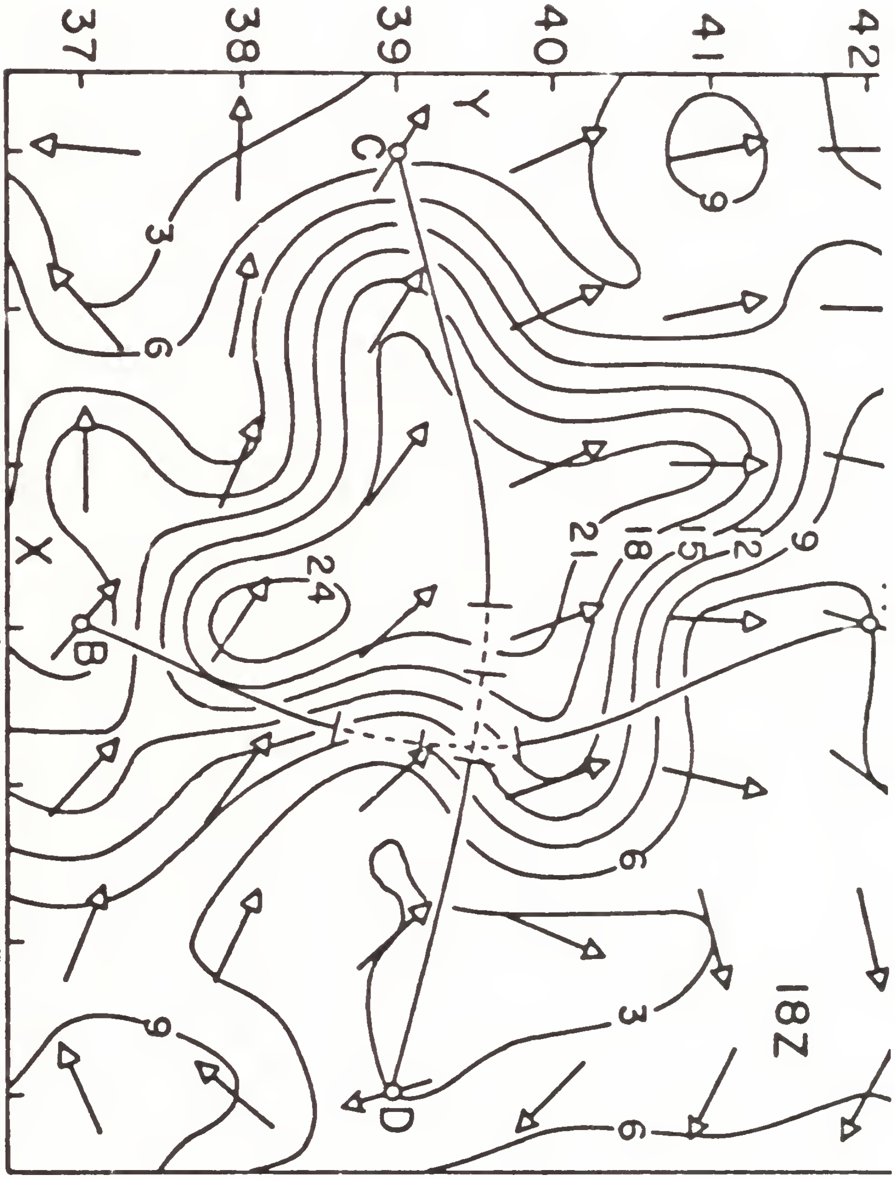
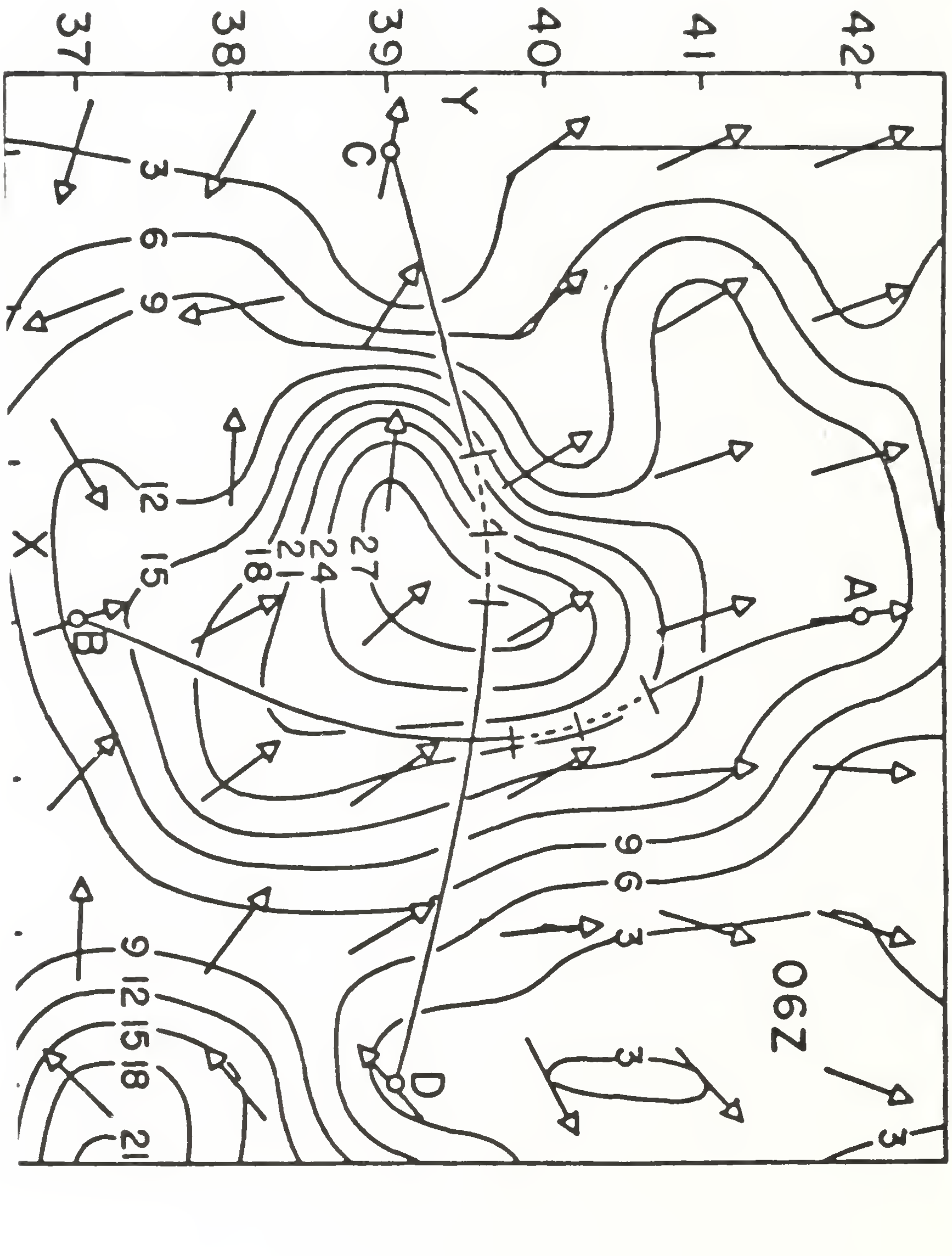


Fig. 5. Sea conditions



ıd 18Z on 5 May 1963.

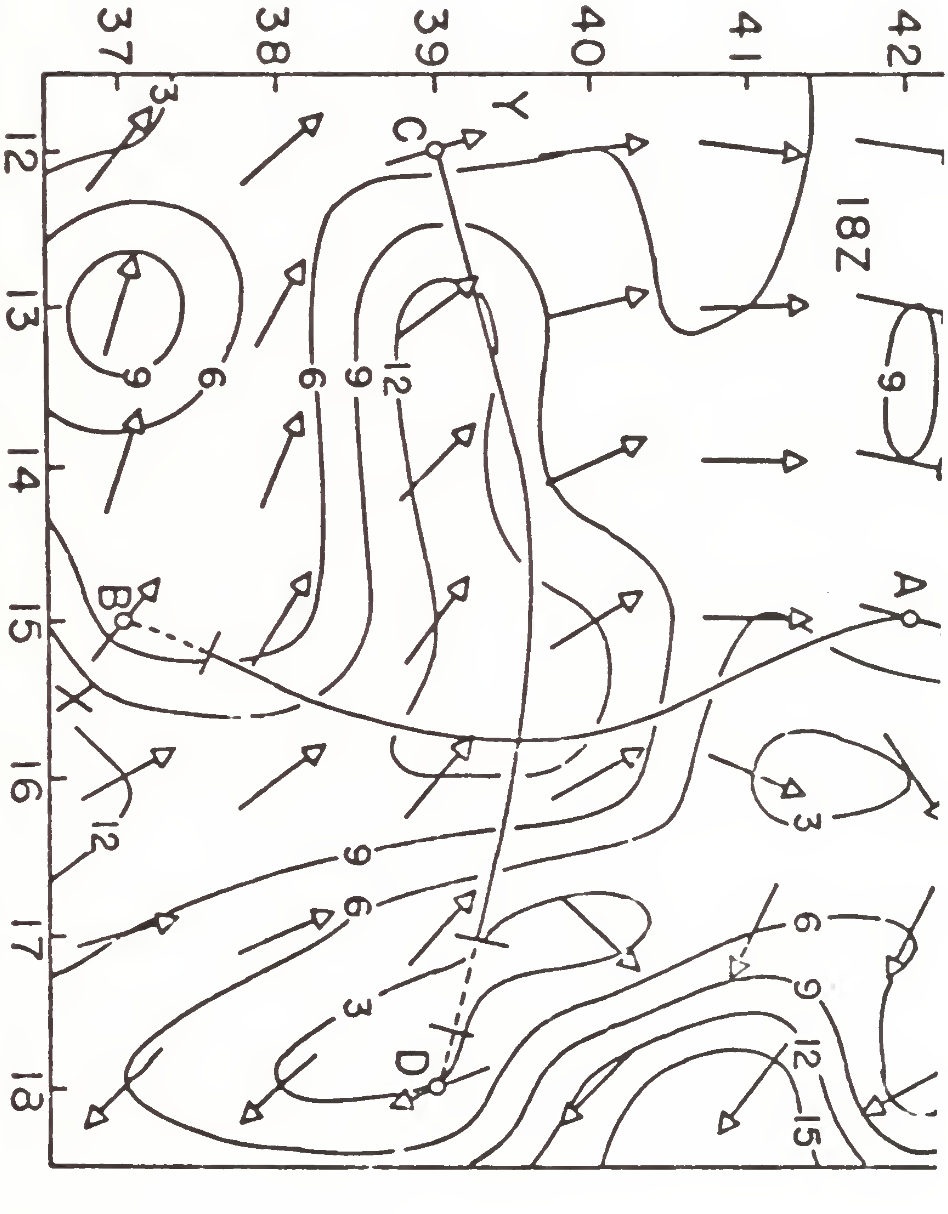
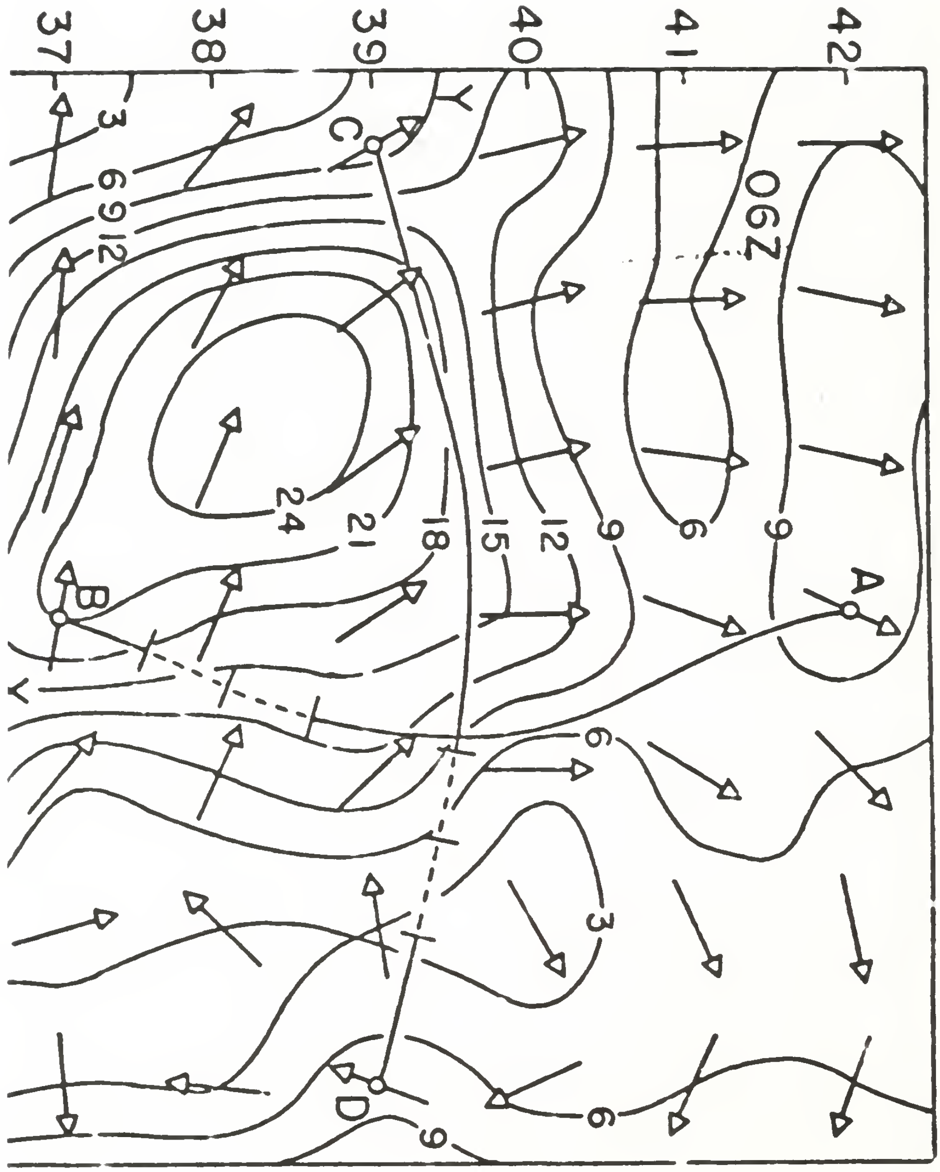


Fig. 6. Sea conditions



ıd 18Z on 6 May 1963.

sary when the wave height was more nearly analytic.

6. Concluding remarks

The numerical integrations involved in the theory of minimal-time ship routing through time-dependent wave fields are found to be feasible. The necessary three-dimensional interpolations in the wave field data, discussed in the Appendix, present no problem. Convergence problems may arise, but can be solved by the described delayed approach to the limit. The authors can supply copies of their Fortran programs for ship routing and for the cubic-interpolation contouring of Figs. 4, 5 and 6.

Acknowledgements. This work was supported by the Office of Naval Research. Aid from the U. S. Navy Fleet Numerical Weather Facility and from the Computer Facility, U. S. Naval Postgraduate School, is acknowledged.

7. Appendix

Some pertinent mathematical details are listed here.

The geographic wave direction ν, measured clockwise from the Nort must be converted to the unit vector $\underline{i}$ cosK+$\underline{j}$ sinK in the stereographic grid system by

$$\begin{aligned} \cos K &= -[(x-31)\cos\nu+(y-31)\sin\nu]/r \\ \sin K &= [(x-31)\sin\nu-(y-31)\cos\nu]/r \end{aligned} \tag{23}$$

where $r^2=(x-31)^2+(y-31)^2$. Then the derivative

$$K_x=\cos K(\partial\sin K/\partial x)-\sin K(\partial\cos K/\partial x). \tag{24}$$

The derivatives $V_x=mv_x+vm_x$ and $V_p=mv_p$ are obtained most conveniently by the implicit differentiation of the equation

$$[v\ \sin(p-K)/b]^2+[(c+v\ \cos(p-K))/a]^2=1 \tag{25}$$

of the elliptical polar velocity diagram, and noting that a,b,c are functions of H(x,y,t), and that K depends on x,y,t. The complexity of the result is reduced by introducing the parameter θ defined by

$$\begin{aligned} \sin\theta &= b\,\sin(q-K)/s = v\,\sin(p-K)/b \\ \cos\theta &= a\,\cos(q-K)/s = [v\,\cos(p-K)+c]/a \end{aligned} \tag{26}$$

where $s^2=a^2\cos^2(q-K)+b^2\sin^2(q-K)$.

The numerical integration of the adjoint Eqs.(7) and (8) demands an interpolation formula for $H(x,y,t)$, $\cos K(x,y,t)$ and $\sin K(x,y,t)$ which guarantees the continuity of these functions and of their first space and time derivatives where any of x,y,t assume grid values. A 16-point interpolation formula to accomplish this is obtained from the 4×4 matrix $\underline{F}$, whose four rows and columns of function entries correspond to four successive x and y grid values respectively. The interpolation mesh cell is the central cell of the array, with x and y measured from the cell center, and with the mesh distance considered to be two units. The formula is

$$F(x,y)=\underline{P}(x)\underline{F}\underline{P}'(y)/256 \tag{27}$$

where the matrix

$$\underline{P}(x)=[(1-x)(x^2-1),(x-1)(3x^2+2x-9),(x+1)(9+2x-3x^2),(x+1)(x^2-1)] \tag{28}$$

and the prime indicates matrix transposition. Interpolation in the time dimension is accomplished by the similar 2-unit-mesh central-difference formula

$$F(t)=[F(-3),F(-1),F(1),F(3)]\underline{P}'(t)/16 \tag{29}$$

which guarantees the continuity of $F(t)$ and dF/dt at each end of the central time interpolation mesh. This formula is consistent with parabolic interpolation at the beginning or end of a time series, where central differences are not available. An interpolated vector $\underline{i}\,\cos K(x,y,t)+\underline{j}\,\sin K(x,y,t)$ should be normalized before use.

REFERENCES

Haltiner, G. J., Hamilton, H. D. and 'Arnason, G., 1962: Minimal-time ship routing. J. Applied Meteor., 1, 1-7.

Faulkner, F. D., 1963: Numerical methods for determining optimum ship routes. Navigation, 10, 351-367.

James, R. W., 1957 (revised 1959): Application of wave forecasts to marine navigation. U. S. Navy Hydrographic Office.

FIGURE LEGENDS

Fig. 1. Polar velocity diagram.

Fig. 2. Ship speed in following, beam and head waves.

Fig. 3. Ship motion in stereographic Oxy plane.

Fig. 4. Sea conditions at 06Z and 18Z on 4 May 1963.

Fig. 5. Sea conditions at 06Z and 18Z on 5 May 1963.

Fig. 6. Sea conditions at 06Z and 18Z on 6 May 1963.

```
..JOB*BLEICK - STAFF BOX 6 - 3 MINUTE EXPRESS - 7/7.5/64
      PROGRAM MINVOY
C   YVARS(1)=XLM1  YVARS(2)=XMU1  YVARS(3)=XLM2  YVARS(4)=XMU2  JJ=0 EW
C   YVARS(5)=X     YVARS(6)=Y     YVARS(7)=XJ    YVARS(8)=S     JJ=1 NS
      DIMENSION KH(19,12,10),HT(19,12,10),KK(19,12,10),COSK(19,12,10),
     + SINK(19,12,10),YVARS(8),DY(8),YC(8),C(4),TAU(300),X(300),Y(300),
     + CAPLAM(300),PP(300),QQ(300),WK(300),VS(300),WH(300),XJ(300),
     + S(300),AK(4,8)
      EQUIVALENCE  (KH,HT), (KK,COSK)
      READ 1, KH,KK,JJ,LMAX,KXST,KYST,KXFN,KYFN,ALF,T,FAC,FMUL
    1 FORMAT ( 60(38I2/), 60(38I2/), 6I3, 4F12.9 )
      PRINT 2, JJ, LMAX, KXST, KYST, KXFN, KYFN, FAC, FMUL
    2 FORMAT (1H0, 6I3, 2F12.9)
      DO 4  I=1,19
      DELI = I-26
      DO 4  J=1,12
      DELJ = J+1
      ROOT = SQRTF(DELI*DELI + DELJ*DELJ)
      DO 4  K=1,10
      HT(I,J,K) = KH(I,J,K)
      ANGLE = KK(I,J,K) * 10
      ANGLE = ANGLE/57.2957 7951
      COS = COSF(ANGLE)
      SIN = SINF(ANGLE)
      COSK(I,J,K) = -(DELI*COS + DELJ*SIN)/ROOT
    4 SINK(I,J,K) =  (DELI*SIN - DELJ*COS)/ROOT
      C(1) = 0.0
      C(2) = 0.5
      C(3) = 0.5
      C(4) = 1.0
      X(1)   = KXST
      Y(1)   = KYST
      TAU(1)= 0.0
      XFIN  = KXFN
      YFIN  = KYFN
      WH(1) = HT(KXST,KYST,1)
      CALL POLAR (COSK(KXST,KYST,1),SINK(KXST,KYST,1),WK(1))
      XSTEP = T/FAC
      CALL ABC (WH(1),AI,BI,CI,DAI,DBI,DCI)
      CAPLAM(1) = 1.0
      S(1) = 0.0
      XJ(1)= 0.0
      DO 18  L=1,LMAX
      Q = ALF
      QQ(1) = Q * 57.2957 7951
      COSA = COSF(ALF)
      SINA = SINF(ALF)
      COSQ = COSA
      SINQ = SINA
      COSQMK= COSQ*COSK(KXST,KYST,1) + SINQ*SINK(KXST,KYST,1)
      SINQMK= SINQ*COSK(KXST,KYST,1) - COSQ*SINK(KXST,KYST,1)
      ABS = AI*COSQMK
      ORD = BI*SINQMK
      HYP = SQRTF(ABS*ABS + ORD*ORD)
      SINB= ORD/HYP
      COSB= ABS/HYP
      VMAJ= AI*COSB - CI
      VMIN= BI*SINB
      VS(1)=SQRTF(VMAJ*VMAJ + VMIN*VMIN)
      CALL POLAR (VMAJ,VMIN,PMK)
      PP(1)= PMK + WK(1)
      XVAR = 0.0
      DO 5   I=1,8
    5 YVARS(I) = 0.0
      YVARS(1) = 1.0
      YVARS(4) = 1.0
      YVARS(5) = X(1)
      YVARS(6) = Y(1)
        N1 = T/XSTEP + 1.0
       XN1 = N1
      STEP = T/XN1
        N2 = N1 + 1
      DO 14  K=2,N2
      DO  7  I=1,4
        XC = XVAR + C(I)*STEP
      DO  6  J=1,8
    6 YC(J)= YVARS(J) + C(I)*AK(I-1,J)
      XLAM = YC(1)*COSA + YC(3)*SINA
      XMU  = YC(2)*COSA + YC(4)*SINA
      CLAM = SQRTF(XLAM*XLAM + XMU*XMU)
      CALL TERP (HT,YC(5),YC(6),XC,H,HX,HY)
```

```
      CALL ABC (H,FA,FB,FC,DA,DB,DC)
      CALL TERP (COSK,YC(5),YC(6),XC,CK,CKX,CKY)
      CALL TERP (SINK,YC(5),YC(6),XC,SK,SKX,SKY)
      ROOT = SQRTF(CK*CK + SK*SK)
        CK = CK/ROOT
        SK = SK/ROOT
       DKX = CK*SKX - SK*CKX
       DKY = CK*SKY - SK*CKY
      COSQ = XLAM/CLAM
      SINQ = XMU/CLAM
      COSQMK = COSQ*CK + SINQ*SK
      SINQMK = SINQ*CK - COSQ*SK
       ABS = FA*COSQMK
       ORD = FC*SINQMK
       HYP = SQRTF(ABS*ABS + ORD*ORD)
      SINB = ORD/HYP
      COSB = ABS/HYP
      VMAJ = FA*COSB - FC
      VMIN = FB*SINB
         V = SQRTF(VMAJ*VMAJ + VMIN*VMIN)
      COSP = (CK*VMAJ - SK*VMIN)/V
      SINP = (SK*VMAJ + CK*VMIN)/V
      COST = VMAJ/V
       VBR = V/18.702 181 818
      DELX = YC(5) - 26.0
      DELY = YC(6) + 1.0
      EMFI = (973.75 + DELX*DELX + DELY*DELY)/1043.638 743
      CAPV = VBR*EMFI
      DY(5)= CAPV*COSP
      DY(6)= CAPV*SINP
      EMFIX= DELX/521.819 3715
      EMFIY= DELY/521.819 3715
      DY(8)= CAPV
      B2MA2= FB*FB - FA*FA
      AMCCB= FA - FC*COSB
       RAT = SINB*(FA*FC + B2MA2*COSB)/(FB*AMCCB)
        VP = RAT * V
      CAPVP= RAT * CAPV
      DIV=1.+RAT**2-(V/AMCCB)**2*((VP*COST/SINB)-(B2MA2*SINB**2/FB))/FB
       DET = YC(1)*YC(4) - YC(2)*YC(3)
      RADIX= SQRTF(1.0 + RAT*RAT)
      DY(7)= (CAPV*DET*DET/DIV) * (RADIX/CLAM)**3
      FNUM = -RAT*AMCCB
      DBDH = DA*COSB**2 + FA*DB*SINB**2/FB - DC*COSB
      RATX = (FNUM*DKX + HX*DBDH)/AMCCB
      RATY = (FNUM*DKY + HY*DBDH)/AMCCB
      VBRX = RATX*VBR
      VBRY = RATY*VBR
      CAPVX= VBRX*EMFI + VBR*EMFIX
      CAPVY= VBRY*EMFI + VBR*EMFIY
      DY(1)= -CAPVX*(YC(1)*COSP + YC(2)*SINP)
      DY(2)= -CAPVY*(YC(1)*COSP + YC(2)*SINP)
      DY(3)= -CAPVX*(YC(3)*COSP + YC(4)*SINP)
      DY(4)= -CAPVY*(YC(3)*COSP + YC(4)*SINP)
      DO 7  J=1,8
    7 AK(I,J) = STEP * DY(J)
      DO 8  J=1,8
    8 YVARS(J) =YVARS(J) + ( AK(1,J)+2.*AK(2,J)+2.*AK(3,J)+AK(4,J) )/6.
      XVAR = XVAR + STEP
      TAU(K) = TAU(K-1) + STEP
      X(K) = YVARS(5)
      Y(K)  = YVARS(6)
      IF (N2-K)  80,81,80
   80 IF (LMAX-L)  82,81,82
   81 XLAM  = YVARS(1)*COSA + YVARS(3)*SINA
       XMU  = YVARS(2)*COSA + YVARS(4)*SINA
      CLAM  = SQRTF(XLAM*XLAM + XMU*XMU)
      CAPLAM(K) = CLAM
      CALL TERP (HT,X(K),Y(K),TAU(K),WH(K),HX,HY)
      CALL ABC (WH(K),FA,FB,FC,DA,DB,DC)
      CALL TERP (COSK,X(K),Y(K),TAU(K),CK,CKX,CKY)
      CALL TERP (SINK,X(K),Y(K),TAU(K),SK,SKX,SKY)
      ROOT = SQRTF(CK*CK + SK*SK)
        CK = CK/ROOT
        SK = SK/ROOT
      COSQ = XLAM/CLAM
      SINQ =  XMU/CLAM
      COSQMK = COSQ*CK + SINQ*SK
      SINQMK = SINQ*CK - COSQ*SK
       ABS = FA*COSQMK
       ORD = FB*SINQMK
```

```
      HYP  = SCRTF(ABS*ABS + ORD*ORD)
      SINB = ORD/HYP
      COSB = ABS/HYP
      VMAJ = FA*COSB - FC
      VMIN = FB*SINB
      VS(K)= SCRTF(VMAJ*VMAJ + VMIN*VMIN)
      COSP = (CK*VMAJ - SK*VMIN)/VS(K)
      SINP = (SK*VMAJ + CK*VMIN)/VS(K)
      CALL POLAR (COSP,SINP,PP(K))
      CALL POLAR (COSQ,SINQ,QQ(K))
      CALL POLAR (CK,SK,WK(K))
      XJ(K) = YVARS(7)/(YVARS(1)*YVARS(4) - YVARS(2)*YVARS(3))
      S(K)  = YVARS(8)
      IF (N2-K)  82,15,82
   82 IF (JJ)  88,83,88
   83 IF (XFIN-X(1))  85,21,84
   84 IF (X(K)-XFIN)  86,11,11
   85 IF (X(K)-XFIN)  11,11,86
   86 IF (Y(K)-3.00)  11,11,87
   87 IF (10.0-Y(K))  11,11,14
   88 IF (YFIN-Y(1))  90,21,89
   89 IF (Y(K)-YFIN)  91,11,11
   90 IF (Y(K)-YFIN)  11,11,91
   91 IF (X(K)-3.00)  11,11,92
   92 IF (16.0-X(K))  11,11,14
   11 N2 = K
   12  T = TAU(K)
      PRINT 13, N2
   13 FORMAT (12H0 EARLY N2= I2/)
      GO TO 81
   14 CONTINUE
   15 PRINT 16
   16 FORMAT (1H06X2HN212X3HALF14X1HT14X1HX13X1HY)
      PRINT 17, N2,ALF,T,X(N2),Y(N2)
   17 FORMAT (I10, 4F15.9 )
      DELX = X(N2) - 26.0
      DELY = Y(N2) + 1.0
      EMFI = (973.75 + DELX*DELX + DELY*DELY)/1043.638 743
      CAPV = VS(N2)*EMFI/18.702 181 818
      XDOT = CAPV*COSP
      YDOT = CAPV*SINP
      FINJ = XJ(N2)
      DETER= FINJ*(XDOT*XLAM + YDOT*XMU)
      DIFX = XFIN - X(N2)
      DIFY = YFIN - Y(N2)
      DIFT = FINJ*(XLAM*DIFX + XMU*DIFY)/DETER
      DIFA = (XDOT*DIFY - YDOT*DIFX)/DETER
         T = T + FMUL*DIFT
       ALF = ALF + FMUL*DIFA
      PRINT 16
      PRINT 17, N2,ALF,T,X(N2),Y(N2)
   18 CONTINUE
      PRINT 19
   19 FORMAT(1H04X3HTAU8X1HX9X1HY6X6HCAPLAM5X2HWH8X2HWK8X2HVS9X1HS8X2HXJ
     +8X2HPP8X2HQQ/)
      PRINT 20,(TAU(I),X(I),Y(I),CAPLAM(I),WH(I),WK(I),
     +          VS(I),S(I),XJ(I),PP(I),QQ(I),   I=1,N2)
   20 FORMAT (11F10.4)
   21 STOP
      END
      SUBROUTINE ABC (H,A,B,C,DA,DB,DC)
      RAD1 = SCRTF((0.038 635 06466*H-0.492 435 9727)*H+2.099 340 872)
      A      = -0.120 197 666*H + 20.748 910 236 - RAD1
      RAD2 = SCRTF((0.032 937 36384*H-0.439 806 7919)*H+1.778 122 940)
      B      = -0.126 879 519*H + 20.633 462 763 - RAD2
      RAD3 = SCRTF((0.010 441 8956*H-0.068 479 3891)*H+0.371 204 6341)
      RAD4 = SCRTF((0.013 051 7910*H-0.193 791 6314)*H+0.814 886 3262)
      C      =  0.125 424 493*H -  0.293 444 8925 - RAD3 + RAD4
      DA     = -0.120 197 666 -(0.038 635 06466*H - 0.246 217 98635)/RAD1
      DB     = -0.126 879 519 -(0.032 937 36384*H - 0.219 903 39595)/RAD2
      DC     =  0.125 424 493 -(0.010 441 89560*H - 0.034 239 69455)/RAD3
     +                        +(0.013 051 79100*H - 0.096 895 81570)/RAD4
      END
      SUBROUTINE POLAR (X,Y,P)
      IF (X)  10,3,10
    3 IF (Y)   8,4,6
    4 P= 0.0
    5 RETURN
    6 P=90.0
    7 RETURN
    8 P= -90.0
```

```
  9 RETURN
 10 P= 57.2957 7951 * ATANF(Y/X)
 11 IF (X)  12,14,14
 12 IF (Y)  15,13,13
 13 P= P + 180.0
 14 RETURN
 15 P= P - 180.0
 16 END
    SUBROUTINE TERP (FUNC, X, Y, T, OUT, CUTX, CUTY)
    DIMENSION FUNC(19,12,10),PT(4,4),P(4),Q(4),PX(4),QY(4),ORD(4),
   +          ORDX(4),ORDY(4),S(4),D(4)
    LL = 1
     L = XINTF(T)
    IF (L)  1,4,1
  1 IF (8-L)  3 2,3
  2 LL = 3
     L = L-2
    GO TO 4
  3 LL = 2
     L = L-1
  4  M = XINTF(X) - 2
     N = XINTF(Y) - 2
    XX = 2.0*(X-INTF(X)) - 1.0
    YY = 2.0*(Y-INTF(Y)) - 1.0
    TT = 2.0*(T-INTF(T)) - 1.0
    XP1= XX + 1.0
    XM1= XX - 1.0
    YP1= YY + 1.0
    YM1= YY - 1.0
    TP1= TT + 1.0
    TM1= TT - 1.0
    X2M= XP1*XM1
    Y2M= YP1*YM1
    T2M= TP1*TM1
    P(1) = -XM1*X2M
    P(2) = ((3.*XX+2.)*XX-9.)*XM1
    P(3) = -2.*XX*XX + 18. - P(2)
    P(4) =  XP1*X2M
    Q(1) = -YM1*Y2M
    Q(2) = ((3.*YY+2. )*YY-9.)*YM1
    Q(3) = -2.*YY*YY + 18. - Q(2)
    Q(4) =  YP1*Y2M
    PX(4)= (3.*XX-1.)*XP1
    PX(1)=  4.*XX - PX(4)
    PX(2)= (9.*XX-11.)*XP1
    PX(3)= -4.*XX - PX(2)
    QY(4)= (3.*YY-1.)*YP1
    QY(1)=  4.*YY - QY(4)
    QY(2)= (9.*YY-11.)*YP1
    QY(3)= -4.*YY - QY(2)
    DO 6  K=1,4
    DO 5  I=1,4
    DO 5  J=1,4
  5 PT(I,J) = FUNC(M+I,N+J,L+K)
    DO 10  I=1,4
    S(I) =  P(1)*PT(1,I)+ P(2)*PT(2,I)+ P(3)*PT(3,I)+ P(4)*PT(4,I)
 10 D(I) =  Q(1)*PT(I,1)+ Q(2)*PT(I,2)+ Q(3)*PT(I,3)+ Q(4)*PT(I,4)
    ORD(K) =(D(1)* P(1) + D(2)* P(2) + D(3)* P(3) + D(4)* P(4))/256.
    ORDX(K)=(D(1)*PX(1) + D(2)*PX(2) + D(3)*PX(3) + D(4)*PX(4))/128.
  6 ORDY(K)=(S(1)*QY(1) + S(2)*QY(2) + S(3)*QY(3) + S(4)*QY(4))/128.
    IF (LL-2)  8,7,9
  7   G = (3.0*TT+2.0)*TT - 9.0
      H = G - 4.0*TT
    OUT =(TM1*(G*ORD( 2)-T2M*ORD( 1))-TP1*(H*ORD( 3)-T2M*ORD( 4)))/16.
    OUTX=(TM1*(G*ORDX(2)-T2M*ORDX(1))-TP1*(H*ORDX(3)-T2M*ORDX(4)))/16.
    OUTY=(TM1*(G*ORDY(2)-T2M*ORDY(1))-TP1*(H*ORDY(3)-T2M*ORDY(4)))/16.
    RETURN
  8 TM3 = TT - 3.0
    OUT = ( TM3*(TM1*ORD( 1)-2.0*TP1*ORD( 2)) + T2M*ORD( 3))/8.0
    OUTX= ( TM3*(TM1*ORDX(1)-2.0*TP1*ORDX(2)) + T2M*ORDX(3))/8.0
    OUTY= ( TM3*(TM1*ORDY(1)-2.0*TP1*ORDY(2)) + T2M*ORDY(3))/8.0
    RETURN
  9 TP3 = TT + 3.0
    OUT = ( TP3*(TP1*ORD( 4)-2.0*TM1*ORD( 3)) + T2M*ORD( 2))/8.0
    OUTX= ( TP3*(TP1*ORDX(4)-2.0*TM1*ORDX(3)) + T2M*ORDX(2))/8.0
    OUTY= ( TP3*(TP1*ORDY(4)-2.0*TM1*ORDY(3)) + T2M*ORDY(2))/8.0
    END
      END
1  6 10 10 10   5-1.362675810 4.97704031150.000000000 0.500000000
C   PROGRAM STRAIGHT
```

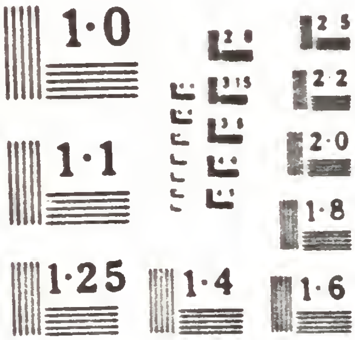
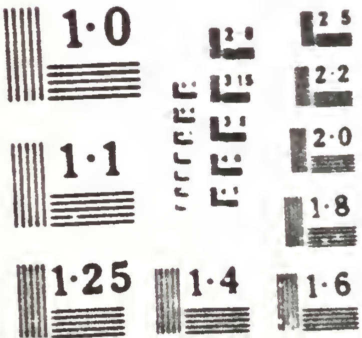
```
C   TO CONVERT PROGRAM MINVOY TO PROGRAM STRAIGHT
C  (1) REPLACE MINVOY CARDS (43-55) BY FOLLOWING CARDS
C      COST  =-SINK(KXST,KYST,1)
C   OR COST  = COSK(KXST,KYST,1)
C      SINT  =-COSK(KXST,KYST,1)
C   OR SINT  =-SINK(KXST,KYST,1)
C      QUAD  = BI*BI*COST*COST + AI*AI*SINT*SINT
C      RADI  = SQRTF(QUAD - (CI*SINT)**2)
C      SINB  = SINT*(AI*RADI-BI*CI*COST)/QUAD
C      COSB  = (BI*COST*RADI+AI*CI*SINT*SINT)/QUAD
C  (2) REPLACE MINVOY CARDS (89-97) BY FOLLOWING CARDS
C      COST  =-SK
C   OR COST  = CK
C      SINT  =-CK
C   OR SINT  =-SK
C      QUAD  = FB*FB*COST*COST + FA*FA*SINT*SINT
C      RADI  = SQRTF(QUAD-(FC*SINT)**2)
C      SINB  = SINT*(FA*RADI - FB*FC*COST)/QUAD
C      COSB  = (FB*COST*RADI + FA*FC*SINT*SINT)/QUAD
C  (3) REPLACE MINVOY CARDS (153-161) BY FOLLOWING CARDS
C      COST  =-SK
C   OR COST  = CK
C      SINT  =-CK
C   OR SINT  =-SK
C      QUAD  = FB*FB*COST*COST + FA*FA*SINT*SINT
C      RADI  = SQRTF(QUAD-(FC*SINT)**2)
C      SINB  = SINT*(FA*RADI - FB*FC*COST)/QUAD
C      COSB  = (FB*COST*RADI + FA*FC*SINT*SINT)/QUAD
C  (4) REPLACE MINVOY CARDS (197-198) BY FOLLOWING CARDS
C      FF  =  (5.00-Y(N2-1))/(Y(N2)-Y(N2-1))
C   OR FF  =  (13.0-X(N2-1))/(X(N2)-X(N2-1))
C        T  = FF*TAU(N2) + (1.0-FF)*TAU(N2-1)

      PROGRAM CONTOUR
      DIMENSION ABS(900),ORD(900),KH(19,12),HT(19,12),KK(19,12),PT(12),
     +COSK(19,12),SINK(19,12),RX(48),RY(48),CON(4,4),IT(12),TI(12),D(3)
      EQUIVALENCE  (IT,TI), (LA,AL)
      READ 1, DS,IHMAX,IDELH,IXMIN,IXMAX,IYMIN,IYMAX,KH,KK,TI,AL
    1 FORMAT ( F4.2, 6I3/, 12(38I2/), 2(10A8/) )
      DO 2  J=1,12
      DO 2  I=1,19
    2 HT(I,J) = KH(I,J)
      PRINT 3, DS, IHMAX, IDELH, IXMIN, IXMAX, IYMIN, IYMAX
    3 FORMAT ( 1H0, F4.2, 6I3 )
      READ 26, (ABS(I), ORD(I), I=1,5)
   26 FORMAT (10F3.1)
      CALL DRAW (5,ABS,ORD,1,0,LA,IT,1.,1.,0,0,2,2,8,8,0,LAST)
      C = DS*DS
      DO 22  IY= IYMIN, IYMAX
       Y = IY
      DO 22  IX= IXMIN, IXMAX
       X = IX
      CALL COEF (HT, IX, IY, PT, CON)
      DO 22  IH= 3, IHMAX, IDELH
       H = IH
      NR = 0
      CALL ROOT (CON, 1, H, D, KER)
      IF (KER)  8,8,4
    4 DO 7  J=1,KER
      ABS(1) = X + (D(J)+1.0)/2.0
      ORD(1) = Y
      IF (NR)  33,35,33
   33 DO 34  I=1,NR
      A = ABS(1) - RX(I)
      B = ORD(1) - RY(I)
      IF (A*A + B*B - C)  7,7,34
   34 CONTINUE
   35 CALL GRAD (PT,ABS(1),ORD(1),DX,DY,QUAD,DELX,DELY,DS,H,0)
      IF (QUAD - 1.E-10)  7,7,5
    5 IF (DELY)  6,36,36
    6 DS  = -DS
      DELX= DELX - DX - DX
      DELY= DELY - DY - DY
      IF (DELY)  7,36,36
   36 ABS(2) = ABS(1) + DELX
      ORD(2) = ORD(1) + DELY
      CALL CUT (NR,ABS,ORD,RX,RY,DS,PT,X,Y,H)
    7 CONTINUE
      IF (KER)  13,13,9
    8 CALL ROOT (CON, 2, H, D, KER)
    9 DO 12  J=1,KER
```

```
      ABS(1) = X
      ORD(1) = Y + (D(J)+1.0)/2.0
      IF (NR)  37,39,37
   37 DO 38  I=1,NR
      A = ABS(1) - RX(I)
      B = ORD(1) - RY(I)
      IF (A*A + B*B - C)  12,12,38
   38 CONTINUE
   39 CALL GRAD (PT,ABS(1),ORD(1),DX,DY,QUAD,DELX,DELY,DS,H,0)
      IF (QUAD - 1.E-10)  12,12,10
   10 IF (DELX)  11,40,40
   11 DS   = -DS
      DELY= DELY - DY - DY
      DELX= DELX - DX - DX
      IF (DELX)  12,40,40
   40 ABS(2) = ABS(1) + DELX
      ORD(2) = ORD(1) + DELY
      CALL CUT (NR,ABS,ORD,RX,RY,DS,PT,X,Y,H)
   12 CONTINUE
   13 CALL ROOT (CON, 3, H, D, KER)
      IF (KER)  18,18,14
   14 DO 17  J=1,KER
      ABS(1) = X + (D(J)+1.0)/2.0
      ORD(1) = Y + 1.0
      IF (NR)  41,43,41
   41 DO 42  I=1,NR
      A = ABS(1) - RX(I)
      B = ORD(1) - RY(I)
      IF (A*A + B*B - C)  17,17,42
   42 CONTINUE
   43 CALL GRAD (PT,ABS(1),ORD(1),DX,DY,QUAD,DELX,DELY,DS,H,0)
      IF (QUAD - 1.E-10)  17,17,15
   15 IF (DELY)  44,44,16
   16 DS   = -DS
      DELX= DELX - DX - DX
      DELY= DELY - DY - DY
      IF (DELY)  44,44,17
   44 ABS(2) = ABS(1) + DELX
      ORD(2) = ORD(1) + DELY
      CALL CUT (NR,ABS,ORD,RX,RY,DS,PT,X,Y,H)
   17 CONTINUE
   18 CALL ROOT (CON, 4, H, D, KER)
      IF (KER)  22,22,19
   19 DO 22  J=1,KER
      ABS(1) = X + 1.0
      ORD(1) = Y + (D(J)+1.0)/2.0
      IF (NR)  45,47,45
   45 DO 46  I=1,NR
      A = ABS(1) - RX(I)
      B = ORD(1) - RY(I)
      IF (A*A + B*B - C)  22,22,46
   46 CONTINUE
   47 CALL GRAD (PT,ABS(1),ORD(1),DX,DY,QUAD,DELX,DELY,DS,H,0)
      IF (QUAD - 1.E-10)  22,22,20
   20 IF (DELX)  48,48,21
   21 DS   = -DS
      DELY= DELY - DY - DY
      DELX= DELX - DX - DX
      IF (DELX)  48,48,22
   48 ABS(2) = ABS(1) + DELX
      ORD(2) = ORD(1) + DELY
      CALL CUT (NR,ABS,ORD,RX,RY,DS,PT,X,Y,H)
   22 CONTINUE
      DO 23  I=1,19
      DELI = I-26
      DO 23  J=1,12
      DELJ = J+1
      RAD  = SQRTF(DELI*DELI + DELJ*DELJ)
      ANGLE = KK(I,J)*10
      ANGLE = ANGLE/57.2957 7951
      COS = COSF(ANGLE)
      SIN = SINF(ANGLE)
      COSK(I,J) = -(DELI*COS + DELJ*SIN)/RAD
   23 SINK(I,J) =   (DELI*SIN - DELJ*COS)/RAD
      IXP = IXMIN + 1
      IYP = IYMIN + 1
      DO 24  IY = IYP, IYMAX
      Y = IY
      DO 24  IX = IXP,IXMAX
      X = IX
      COS = COSK(IX,IY)
```

```
      SIN = SINK(IX,IY)
      ABS(1) = X + COS/3.0
      ORD(1) = Y + SIN/3.0
      ABS(2) = X + X - ABS(1)
      ORD(2) = Y + Y - ORD(1)
      ABS(3) = ABS(2) + (1.732050807*COS - SIN)/18.0
      ORD(3) = ORD(2) + (1.732050807*SIN + COS)/18.0
      ABS(4) = ABS(3) + SIN/9.0
      ORD(4) = ORD(3) - COS/9.0
      ABS(5) = ABS(2)
      ORD(5) = ORD(2)
      DO 28  I=1,5
      ABS(I) = 0.8*ABS(I) - 4.0
   28 ORD(I) = 0.8*ORD(I) - 3.2
      CALL CRAW (5,ABS,ORD,2,0,LA,IT,1.,1.,0,0,2,2,8,8,0,LAST)
   24 CONTINUE
      READ 29, (ABS(I), I=1,52)
   29 FORMAT ( 4(13F6.3/) )
      READ 29, (ORD(I), I=1,52)
      DO 30  I=1,52
      ABS(I) = 0.8*ABS(I) - 4.0
   30 ORD(I) = 0.8*ORD(I) - 3.2
      CALL CRAW (52,ABS,ORD,2,0,LA,IT,1.,1.,0,0,2,2,8,8,0,LAST)
      READ 29, (ABS(I), I=1,52)
      READ 29, (ORD(I), I=1,52)
      DO 32  I=1,52
      ABS(I) = 0.8*ABS(I) - 4.0
   32 ORD(I) = 0.8*ORD(I) - 3.2
      CALL CRAW (52,ABS,ORD,3,0,LA,IT,1.,1.,0,0,2,2,8,8,0,LAST)
   25 STOP 25
      END
      SUBROUTINE COEF (HT, IX, IY, P, CON)
      DIMENSION  HT(19,12), P(12), CON(4,4)
      P(1) = HT(IX,IY-1)
      P(2) = HT(IX+1,IY-1)
      LX = IX-2
      DO 1  I=1,4
      P(2+I)= HT(LX+I,IY)
    1 P(6+I)= HT(LX+I,IY+1)
      P(11) = HT(IX,IY+2)
      P(12) = HT(IX+1,IY+2)
      CON(1,1) = P(6) -P(3)+ 3.0*(P(4)-P(5))
      CON(2,1) = P(3) - P(4) - P(5) + P(6)
      CON(3,1) = P(3)-P(6) + 11.*(P(5)-P(4))
      CON(4,1) =-P(3)-P(6) + 9.0*(P(5)+P(4))
      CON(1,3) = P(10)-P(7)+ 3.0*(P(8)-P(9))
      CON(2,3) = P(10)- P(9) - P(8) + P(7)
      CON(3,3) = P(7)-P(10)+ 11.*(P(9)-P(8))
      CON(4,3) =-P(7)-P(10)+ 9.0*(P(9)+P(8))
      CON(1,2) = P(11)-P(1)+ 3.0*(P(4)-P(8))
      CON(2,2) = P(1) - P(4) - P(8) + P(11)
      CON(3,2) = P(1)-P(11)+ 11.*(P(8)-P(4))
      CON(4,2) =-P(1)-P(11)+ 9.0*(P(8)+P(4))
      CON(1,4) = P(12)-P(2)+ 3.0*(P(5)-P(9))
      CON(2,4) = P(12)- P(9) - P(5) + P(2)
      CON(3,4) = P(2)-P(12)+ 11.*(P(9)-P(5))
      CON(4,4) =-P(2)-P(12)+ 9.0*(P(9)+P(5))
      END
      SUBROUTINE ROOT (CON, KS, H, D, KER)
      DIMENSION  CON(4,4), D(3), X(3)
      FOURTH = CON(4,KS) - 16.0*H
      IF (CON(1,KS))  12,2,12
    2 IF (CON(2,KS))   6,3, 6
    3 IF (CON(3,KS))  11,4,11
    4 IF (FOURTH)      9,5, 9
    5 D(1) = -1.0
      D(2) =  1.0
      KER  = 2
      RETURN
    6 B = 0.5*CON(3,KS)/CON(2,KS)
      C = FOURTH/CON(2,KS)
      QUAD = B*B - C
      RAD  = SQRTF(ABSF(QUAD))
      IF (QUAD)  7,8,10
    7 IF (RAD - 1.E-5)  8,8,9
    8 X(1) = -B
      KER  = 1
      GO TO 28
    9 KER  = 0
      RETURN
   10 X(1) = -B + RAD
```

```
      X(2) = -B - RAD
      KER  = 2
      GO TO 28
   11 X(1) = -FOURTH/CON(3,KS)
      KER  = 1
      GO TO 28
   12 A = CON(2,KS)/CON(1,KS)
      B = CON(3,KS)/CON(1,KS)
      C =      FOURTH/CON(1,KS)
      P = A*A/9.0 - B/3.0
      Q = A*B/6.0 - C/2.0 - A*A*A/27.0
      RAD = SQRTF(ABSF(P))
      IF (P)  13,16,18
   13 PHI3= ASINH(Q/(RAD*RAD*RAD))/3.0
      X(1) = 2.0*RAD*SINH(PHI3) - A/3.0
      KER  = 1
      IF (1.732050807*RAD*COSH(PHI3) - 1.E-5)  15,15,28
   15 X(2) = -(X(1)+A)/2.0
      KER  = 2
      GO TO 28
   16 X(1) = (2.0*ABSF(Q))**.33333333333333 - A/3.0
      KER  = 1
      IF (Q)  17,28,28
   17 X(1) = -X(1) - A/1.5
      GO TO 28
   18 ARG  = Q/(RAD*RAD*RAD)
      IF (ABSF(ARG) - 1.0)  27,25,19
   19 PHI3 = ACOSH(ABSF(ARG))/3.0
      IF (Q)  21,24,24
   21 X(1) = -2.0*RAD*COSH(PHI3) - A/3.0
   22 KER  =  1
      IF (1.732050807*RAD*SINH(PHI3) - 1.E-5)  15,15,28
   24 X(1) = 2.0*RAD*COSH(PHI3) - A/3.0
      GO TO 22
   25 X(1) = 2.0*ABSF(Q)**.33333333333333 - A/3.0
      X(2) = -(X(1)+A)/2.0
      KER  = 2
      IF (Q)  26,28,28
   26 X(2) = -X(2) - A/1.5
      GO TO 17
   27 PHI3 = ACOSF(ARG)/3.0
      SIN  = 1.732050807*RAD*SINF(PHI3)
        Z  = RAD*COSF(PHI3)
      X(1) = Z + Z - A/3.0
      X(2) = -Z + SIN - A/3.0
      X(3) = X(2) - SIN - SIN
      KER  = 3
   28 JAR = 0
      DO 30  I=1,KER
      IF (ABSF(X(I)) - 1.0)  29,29,30
   29 JAR = JAR + 1
      D(JAR) = X(I)
   30 CONTINUE
      KER = JAR
      END
      SUBROUTINE GRAD (P, X, Y, DX, DY, QUAD, DELX, DELY, DS, H, KD)
      DIMENSION   P(12)
      XX = 2.0*(X - INTF(X)) - 1.0
      YY = 2.0*(Y - INTF(Y)) - 1.0
      XP1= XX + 1.0
      XM1= XX - 1.0
      YP1= YY + 1.0
      YM1= YY - 1.0
       TX= 2.0*XX
       TY= 2.0*YY
      CIR =3.0*(XX*XX+YY*YY) - 10.0
      X2M = XP1*XM1
      Y2M = YP1*YM1
      SXP = 6.0*XX + 2.0
      SXM = SXP - 4.0
      SYP = 6.0*YY + 2.0
      SYM = SYP - 4.0
      QXP = 18.0*XX + 2.0
      QXM = QXP - 4.0
      QYP = 18.0*YY + 2.0
      QYM = QYP - 4.0
       Q1 = X2M*P(10) + Y2M*P(12) - (CIR-TX-TY)*P(9)
       Q2 = X2M*P( 7) + Y2M*P(11) - (CIR+TX-TY)*P(8)
       Q3 = X2M*P( 3) + Y2M*P( 1) - (CIR+TX+TY)*P(4)
       Q4 = X2M*P( 6) + Y2M*P( 2) - (CIR-TX+TY)*P(5)
      P1 = (TX*P(10) - SXM*P(9))*XP1
```

```
      P2 = (TX*P( 7) - SXP*P(8))*XM1
      P3 = (TX*P( 3) - SXP*P(4))*XM1
      P4 = (TX*P( 6) - SXM*P(5))*XP1
      P5 = (TY*P(12) - SYM*P(9))*YP1
      P6 = (TY*P( 2) - SYP*P(5))*YM1
      P7 = (TY*P( 1) - SYP*P(4))*YM1
      P8 = (TY*P(11) - SYM*P(8))*YP1
      HX  = (YP1*(Q1-Q2+P1-P2) + YM1*(Q3-Q4+P3-P4))/16.0
      HY  = (XP1*(Q1-Q4+P5-P6) + XM1*(Q3-Q2+P7-P8))/16.0
      QUAD = HX*HX + HY*HY
      IF (QUAD - 1.E-10)  3,3,2
    2 IF (KC)  5,4,5
    4 HXX = (YP1*(SXP*P(10)-QXP*P(9)-SXM*P(7)+QXM*P(8)) +
     +         YM1*(SXM*P( 3)-QXM*P(4)-SXP*P(6)+QXP*P(5)))/8.0
      HYY = (XP1*(SYP*P(12)-QYP*P(9)-SYM*P(2)+QYM*P(5)) +
     +         XM1*(SYM*P( 1)-QYM*P(4)-SYP*P(11)+QYP*R(8)))/8.0
      HXY = (P1-P4+P3-P2+P5-P8+P7-P6+Q1-Q2+Q3-Q4)/4,0
      RAD = SQRTF(QUAD)
      DX  =  DS*HY/RAD
      DY  = -DS*HX/RAD
      SEC = 0.5*(HXX*DX*DX + HXY*DX*DY + HYY*DY*DY)/QUAD
      DELX= DX - HX*SEC
      DELY= DY - HY*SEC
      RETURN
    5 DH  = H - (YP1*(XP1*Q1-XM1*Q2) + YM1*(XM1*Q3-XP1*Q4))/32.0
      DX = CH*HX/QUAD
      DY = CH*HY/QUAD
      DELX = DH*DH/QUAD
    3 END
      SUBROUTINE OUT (NR,ABS,ORD,RX,RY,DS,PT,X,Y,H)
      DIMENSION  ABS(900),ORD(900),RX(48),RY(48),PT(12)
      C = 2.0*DS*DS
      NPT = 1
      NR  = NR + 1
      RX(NR) = ABS(1)
      RY(NR) = ORD(1)
      GO TO 1
    4 IF (40-NPT)  20,20,24
   24 CALL GRAD (PT,ABS(NPT),ORD(NPT),DX,DY,QUAD,DELX,DELY,DS,H,0)
      IF (QUAD - 1.E-10)  19,19,5
    5 ABS(NPT+1) = ABS(NPT) + DELX
      ORD(NPT+1) = ORD(NPT) + DELY
    1 CALL GRAD(PT,ABS(NPT+1),ORD(NPT+1),DX,DY,QUAD,DELX,DELY,DS,H,1)
      IF (QUAD - 1.E-10)  19,19,6
    6 IF (DELX - C)  25,25,19
   25 NPT = NPT + 1
      ABS(NPT) = ABS(NPT) + DX
      ORD(NPT) = ORD(NPT) + DY
      NPT1= NPT
      IF (ORD(NPT)-Y)  7,10,8
    7 NPT1= NPT1 - 1
      GO TO 10
    8 IF (Y+1.0-ORD(NPT))  9,10,10
    9 NPT1= NPT1 - 1
   10 IF (ABS(NPT1)-X)  11,14,12
   11 NPT1= NPT1 - 1
      GO TO 15
   12 IF (X+1.0-ABS(NPT1))  11,14,14
   14 IF (NPT-NPT1)  15,4,15
   15 NPT = NPT1
   19 IF (NPT-1)  26,26,20
   20 NR = NR + 1
      RX(NR) = ABS(NPT)
      RY(NR) = ORD(NPT)
      DO 21  I=1,NPT
      ABS(I) = 0.8*ABS(I) - 4.0
   21 ORD(I) = 0.8*ORD(I) - 3.2
      LA = 4H
      CALL CRAW (NPT,ABS,ORD,2,0,LA,7,1.,1.,0,0,2,2,8,8,0,LAST)
      RETURN
   26 NR = NR - 1
      END
      END
,10 21  3  6 13  4 10 ZEROTH
```


1·0
1·1
1·25
1·4
2·5
2·2
2·0
1·8
1·6


1·0
1·1
1·25
1·4
2·5
2·2
2·0
1·8
1·6

DEFENSE TECHNICAL INFORMATION CENTER
CAMERON STATION
ALEXANDRIA, VIRGINIA 22304-6145

OFFICIAL BUSINESS - PENALTY FOR PRIVATE USE, $300
POSTMASTER: DO NOT FORWARD

AD Number	Class	Pages	Type Copy	Source	Priority
604353	U	0	H	F	

Received Date: 14-SEP-95

35

To [redacted]

Requested By : JODY OTTILIGE

ATTENTION : DR M RENEKER CODE 524 -LIB DIR

NAVAL POSTGRADUATE SCH

MONTEREY, CA 93943-5000

3092880001